Esercizio di Sincronizzazione Tra Processi: Moltiplicazione di Matrici (esame 12 Giugno 1996)

Testo:

Date de matrici A e B di dimensioni N*N se ne vuole calcolare il prodotto ed inserire il risultato in B (B=A*B) affidando il calcolo ad N*N processi concorrenti ognuno dei quali dedito al calcolo di un singolo elemento della matrice. Commentando adeguatamente il codice si gestisca la politica di sincronizzazione dei processi attraverso un monitor che sia in grado di evitare che un processo sovrascriva il valore della matrice B prima che tutti i processi che hanno bisogno di tale valore per effettuare il calcolo lo abbiano letto. Si faccia uso di pseudo-codice simbolico per evidenziare le parti di lettura, scrittura e calcolo della matrice, implementando effettivamente solo la parte di codice relativa alla sincronizzazione tra i processi.
Si ricorda agli smemorati che: B=A*B → b(i,j)=a(i,1)*b(1,j) + a(i,2)*b(2,j) + ... + a(i,N)*b(N,j)

Soluzione:

La moltiplicazione A*B pone il seguente problema di sincronizzazione: i processi dediti al calcolo di una cella (i, j) devono aspettare, prima di poter scrivere il nuovo valore di tale cella nella matrice B, che tutti gli N processi dediti al calcolo degli elementi della colonna j-esima abbiano letto gli elementi di tale colonna nella matrice B. Il problema si risolve perciò attraverso un monitor dotato di due procedure:
dopo aver letto gli elementi della matrice, il processo (i, j) chiama la prima procedura per notificare al monitor tale lettura e, se esso è l'ultimo ad aver letto la colonna j, per svegliare gli altri processi che si erano fermati ad aspettare di scrivere un elemento su tale colonna;
dopo aver calcolato il valore della cella (i, j) il processo, prima di effettuare la scrittura, chiama la seconda procedura del monitor per controllare che tutti abbiano già letto la colonna j e , in caso contrario, per sospendersi.
Per realizzare queste funzionalità bastano un array di N variabili condition, una per ogni colonna, e un array di interi che tenga conto, per ogni colonna, di quanti processi la hanno già letta.
Si fa rilevare che poiché le letture possono essere effettuate in concorrenza dai processi e poiché ogni processo è il solo a riscrivere una certa cella di B non è necessario porre in mutua esclusione queste fasi.
Queste considerazioni, che già di per sé potrebbero essere considerate una soluzione accettabile dell'esercizio, si traducono nel codice pseudo-C che segue (per ribadire che la sintassi adottata, purché chiara, è inessenziale ai fini del giudizio in sede d'esame, sia essa C, Pascal, Fortran o Basic).

```
#define N un_qualche_numero

int A[N][N], B[N][N];


process calcola(int i, int j)
{       < leggi riga i di A e colonna j di B >
        controller.notifica_lettura(j);
        < calcola elemento (i, j) di B >
        controller.permesso_scrittura(j);
        < scrivi elemento (i, j) di B >         }


monitor controller()
{       condition aspetta[N];  /* array di N variabili condition */
        int quanti[N] = 0;      /* tutti gli elementi della matrice inizializzati a zero */

        procedure entry notifica_lettura (int j)
        {       quanti[j]++;
                if (quanti[j] == N)
                        while (aspetta[j].queue) aspetta[j].signal;     }

        procedure entry permesso_scrittura (int j)
        {       if (quanti[j] < N)
                        aspetta[j].wait;        }
}
```